In step **810**, a control server **100** may identify a software download to be provided to a plurality of secure processors **200***a*-**200***n*. As discussed above, the software download may represent a new software application or an updated version of an existing application currently loaded on the secure processors **200***a*-**2**-**00***n*. In step **820**, the control server **100** generates an encryption key and encrypts the application code image using the encryption key. Thus, unlike previous examples described herein in which an application code image is encrypted multiple times with different encryption keys corresponding to different secure processors **200***a*-**200***n*, in this example the application code image may be encrypted a single time only using a common encryption key.

In step **830**, the control server **100** retrieves the personalized unit data for each of the secure processors **200***a*-**200***n* that will be provided the software. As described in the above examples, this personalized unit data (e.g., seed values, download sequence numbers, etc.) may be stored by the control server **100** in the processor information database **104**. In step **840**, the control server **100** generates a unique encryption key for each of the recipient secure processors **200***a*-**200***n*. This step may be similar (or identical) to step **330** and other examples in which unique encryption keys are generated for different secure processors **200***a*-**200***n*.

In step **850**, the control server **100** uses the unique encryption keys generated in step **840** to separately encrypt the common decryption key that may be used for decrypting the application code image. In step **860**, the control server **100** transmits the separately encrypted common decryption key to each of the secure processors **200***a*-**200***n*. Thus, in this example, each of the secure processors **200***a*-**200***n* will receive the same common decryption key for decrypting the application code image. However, the common decryption key will be separately encrypted for each of the secure processors **200***a*-**200***n* using the techniques described above. In this example, a secure processor **200** may receive and decrypt the uniquely encrypted common decryption key intended for that secure processor, whereas none of the other secure processors **200***a*-**200***n* would be able to decrypt the common decryption key after it was encrypted specifically for secure processor **200**. Each secure processor **200***a*-**200***n* would thus use its own personalized unit data **204** and would generate its own decryption key in order to decrypt the common decryption key for the application code image.

In step **870**, the control server **100** transmits the same encrypted code image to each of the recipient secure processors **200***a*-**200***n*. Each secure processor **200***a*-**200***n*, having separately received and uniquely decrypted the common decryption key, then may decrypt and load the application code image using the common decryption key.

The example illustrated in FIG. **8** incorporates various techniques and features described herein, while also permitting a control server **100** to encrypt and transmit a single encrypted application code image to a plurality of secure processors **200***a*-**200***n* or other computing devices. Potential advantages may result from this technique and similar examples when the size of an application code image is large and the time/costs associated with encrypting and/or transmitting the application code image are significant. For example, when transmitting very large application code images, or when using very time-consuming encryption algorithms to encrypt application code images, it may be potentially beneficial to use the unique one-time encryption and decryption keys only to encrypt/decrypt the common decryption key, and not the application code image itself.

Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

We claim:

1. A method comprising:

responsive to receiving first data corresponding to a software download, retrieving, by a secure device, personalized unit data and a sequence number stored in a protected memory of the secure device;

iteratively executing, by the secure device, a decryption key generation algorithm a number of times based on the sequence number to generate a first decryption key based on the personalized unit data and the sequence number;

decrypting, by the secure device, the first data using the first decryption key; and

in response to determining, by the secure device, that the decryption of the first data was successful, that the software download was successfully executed by the secure device, or both:

transmitting a confirmation message to a control server for incrementing or decrementing a sequence number associated with the secure device at the control server.

2. The method of claim **1**, further comprising:

in response to determining that the decryption of the first data using the first decryption key was successful,

loading the first data into an executable memory of the secure device.

3. The method of claim **1**, further comprising:

receiving second data;

in response to receiving the second data, generating a second decryption key based on the personalized unit data and the incremented or decremented sequence number, the generating comprising iteratively executing the decryption key generation algorithm a number of times based on the incremented or decremented sequence number, wherein the second decryption key is different than the first decryption key; and

decrypting the second data using the second decryption key.

4. The method of claim **3**, wherein generating the first decryption key comprises iteratively generating a predetermined number (N) of decryption keys using a decryption key generation algorithm and using the Nth generated decryption key as the first decryption key.

5. The method of claim **4**, wherein generating the second decryption key comprises iteratively generating N−1 decryption keys using the decryption key generation algorithm and using the Nth−1 generated decryption key as the second decryption key.

6. The method of claim **3**, wherein the first data corresponds to a first software code image, and wherein the second data corresponds to a second software code image.

7. The method of claim **1**, wherein generating the first decryption key comprises using a Secure Hash Algorithm (SHA).

8. The method of claim **1**, wherein the personalized unit data is based on a serial number of the secure device.

9. The method of claim **1**, wherein the determining, by the secure device, comprises determining that the decryption of the first data using the first decryption key was successful.

10. The method of claim **1**, wherein the

determining, by the secure device, comprises determining that the software download was successfully executed by the secure device.